

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:	§	
Deolasee, et al.	§	
	§	Group Art Unit: 2114
Serial No.: 10/812,139	§	
	§	
Confirmation No.: 2165	§	
	§	Examiner: Bonura, Timothy M.
Filed: March 29, 2004	§	
	§	
For: METHOD AND APPARATUS FOR PERFORMING BACKUP STORAGE OF CHECKPOINT DATA WITHIN A SERVER CLUSTER	§ § § §	

MAIL STOP APPEAL BRIEF-PATENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

APPEAL BRIEF

Appellants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner dated August 13, 2007, finally rejecting claims 1-10 and 12-19.

TABLE OF CONTENTS

1.	Identification Page.....	1
2.	Table of Contents	2
3.	Real Party in Interest	3
4.	Related Appeals and Interferences	3
5.	Status of Claims	3
6.	Status of Amendments	3
7.	Summary of Claimed Subject Matter	4
8.	Grounds of Rejection to be Reviewed on Appeal	5
9.	Arguments	6
10.	Conclusion	13
11.	Claims Appendix	14
12.	Evidence Appendix	17
13.	Related Proceedings Appendix	18

Real Party in Interest

The real party in interest is Symantec Operating Corporation, located in Cupertino, California.

Related Appeals and Interferences

Appellants assert that no appeals or interferences regarding related claims are known to the Appellant, Appellants' legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-19 are pending in the application. Claims 1-10 and 12-19 stand finally rejected as discussed below. The final rejection of claims 1-10 and 12-19 is appealed.

Status of Amendments

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments of the invention provide a method and apparatus for performing backup storage of checkpoint data within a server cluster (p.2, paragraph [0007]).

In the embodiments of independent claim 1, a method for providing fault tolerant checkpoint data within a server cluster having a production server and a plurality of backup servers comprises (p.4, paragraph [0014]; FIG. 1): accessing checkpoint data within the production server (p.5, paragraph [0014] and p. 6 paragraph [0023]; step 204, FIG. 2); distributing the checkpoint data (p.5, paragraph [0014]; steps 206-212, FIG. 2); and storing the distributed checkpoint data on the plurality of backup servers (p.5, paragraph [0019]; step 214, FIG. 2).

In the embodiments of independent claim 9, a system for providing fault tolerant checkpoint data comprises (p.4, paragraph [0014]; FIG. 1): a production server for generating checkpoint data (p.5, paragraph [0019]; 108, FIG. 1); means for forming distributed checkpoint data comprising subsegments of the checkpoint data (p. 5, paragraph [0014]; 122 and 136, FIG. 1); and a plurality of backup servers for storing the distributed checkpoint data, where each of the backup servers in said plurality of backup servers stores at least one subsegment of the distributed checkpoint data (p. 5, paragraphs [0014] and [0019] – [0020]; 110, FIG. 1).

In the embodiments of independent claim 14, an apparatus for generating fault tolerant checkpoint data comprises (p. 4, paragraph [0014]; FIG. 1): a first server (108, FIG. 1) that accesses checkpoint data (p. 4, paragraph [0014]), segments the checkpoint data (p. 5, paragraphs [0014] and [0019] – [0020]), and supplies the segments of checkpoint data to a plurality of second servers (110, FIG. 1) for providing fault tolerant checkpoint data (p. 6, paragraph [0023]).

In the embodiments of independent claim 17, a method of generating fault tolerant checkpoint data comprises (p. 4, paragraph [0014]; FIG. 2): accessing checkpoint data that is produced by a first server (p. 5, paragraph [0014] – [0020]; step 204, FIG. 2) segmenting the checkpoint data (p. 6, paragraph [0024]; step 208, FIG. 2); and supplying the segments of checkpoint data to a plurality of second servers (p. 6, paragraph [0024] to p. 7, paragraph [0025]; step 212, FIG. 2).

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1-10 and 12-19 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over U.S. Patent No. 5,734,814 issued to Corbin et al. ("*Corbin*").

ARGUMENTS

I. THE EXAMINER ERRED IN REJECTING CLAIMS 1-10 AND 12-19 UNDER 102(e) BECAUSE CORBIN FAILS TO TEACH ALL THE ELEMENTS OF EACH OF THE CLAIMS.

A. Anticipation of Claims 1-2 and 7-8 over Corbin

Claims 1-2 and 7-8 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over *Corbin* on grounds that *Corbin* teaches the claimed subject matter therein.

Corbin generally teaches a computing system utilizing redundant storage devices arranged in the RAID disk array, data is stored in the computing system using a memory cache created from system memory and the disk memory. See *Corbin*, at abstract. A checkpoint module detects a fault in the computing system and generates a fault indication, and a cache manager writes data and parity to the memory cache in a first mode, and writes data and parity to the storage device in a second mode. *Id.* In response to the fault indication, the checkpoint module copies the data contained in the cache to the disk array, and switches the cache manager from the first mode to the second mode. *Id.*

In contrast, the Appellants' invention, as recited in claim 1, is a method for providing fault tolerant checkpoint data within a server cluster having a production server and a plurality of backup servers comprising accessing checkpoint data within the production server, distributing the checkpoint data and storing the distributed checkpoint data on the plurality of backup servers.

"Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 221 USPQ 481, 485 (Fed. Cir. 1984). Since *Corbin* does not teach each and every element of independent claim 1, *Corbin* does not anticipate Appellants' invention recited in claim 1.

First, *Corbin* does not teach or suggest the claimed limitation *checkpoint data* of the Appellants' invention as recited in independent claim 1. The Examiner incorrectly cited *Corbin*, col. 2, lines 1-3 as a disclosure of the claimed limitation *checkpoint data*.

The cited portion refers to a cache for storing parity and data information but not checkpoint data. The cache in *Corbin* stores “old” and/or “new” data as well as old and/or new parity related to the old and/or new data, respectively. Notably, the “old” and “new” dichotomy refers to the data and parity currently synchronized on the RAID-5 disk array and a set of new data writes to the RAID-5 configuration requiring synchronization, respectively. See *Corbin*, col. 1, lines 43-57 and col. 2 lines 47-53. In fact, *Corbin* refers to writing the new data and the new parity to the final locations on the disk [RAID-5 configuration]. See *Corbin*, col. 5, lines 38-40 and col. 6 lines 56-59. Moreover, the system in *Corbin* operates in response to a request to write the new data to the disk array. See *Corbin*, Fig.4, col. 5, lines 25-26. In addition, *Corbin* teaches invalidation of the new data and parity in the cache after completing the set of writes of the new data to the RAID-5 disk array. See, *Corbin*, col. 5, lines 60-66 (“This information is marked as no longer being usage for data recovery purposes”).

Checkpoint data, however, is essentially the data being used by a specific application at a particular moment in time. See Present Specification, pg. 1, paragraph [0003]. The checkpoint data (and subsegments thereof) is associated with the state of the production server and the specific application using such checkpoint data at that particular moment in time. See Present Specification, pg. 5, paragraph [0019]. Whereas, the data in the cache of *Corbin* does not include information regarding the state of the production server nor does it relate the state to any specific application at a particular moment in time. Furthermore, some of the data in the cache in *Corbin* is invalidated. Checkpoint data, on the other hand, is not invalidated after distribution and can be used in subsequent data recovery tasks. Hence, checkpoint data is not the same as a set of new writes to the RAID-5 disk array. Accordingly, *Corbin* does not teach or suggest the claimed limitation *checkpoint data*.

Second, *Corbin* is devoid of any teaching or suggestion of the claimed limitations *distributing the checkpoint data and storing the distributed checkpoint data on the plurality of backup servers*. The Examiner has confused the writing of new data to the RAID-5 disk array as taught in *Corbin* with distributing and storing the checkpoint data on the plurality of backup servers as claimed by the Appellants’. The Examiner has not provided any specific teaching in *Corbin* that describes or discloses the distribution and

storage of checkpoint data from the production server amongst a plurality of backup servers. Instead, the portion of *Corbin* cited by the Examiner (col. 1, lines 16-22) teaches copying the contents of the cache to the drives of the RAID-5 disk array, but not the plurality of backup servers. See Final Office Action, pg. 2 and Response to Final Office Action pg. 2-3.

Notably, the plurality of backup server differs from the RAID-5 disk array in numerous ways. A disk drive in the RAID-5 disk array is not equivalent to a backup server and cannot operate as such. See Response to Non-Final Office Action, pg. 6 and Response to Final Office Action, pg 6. For example, the backup server can resume execution of an application if the production server fails, but the disk drive, by itself, cannot perform such a function. *Id.* In fact, *Corbin* admits that a failure at the computing system requires rebooting and replacement of the failed component. See *Corbin*, col. 7, lines 4-19. Thus, an application cannot resume execution after a failure of the system according to the teachings of *Corbin*. The Examiner also stated that *Corbin* shows the RAID-5 configuration as having multiple host-client relationships, where the Examiner considered the “host-client relationships” to be multiple servers. (Final Office Action, pp. 5-6). A “relationship”, however, is not a server.

Third, the Examiner has impermissibly argued that the claimed limitations *distributing the checkpoint data and storing the distributed checkpoint data on the plurality of backup servers* should be afforded no patentable weight because a recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and *Corbin*. As stated in previous responses, the above claimed limitations are not intended uses of the method as claimed by the Appellants, but actions performed in order to provide fault-tolerant checkpoint data within a server cluster. See Response to Final Office Action, pgs 6-7.

Additionally, there is no requirement that independent claim 1, a process claim, recite any “structural differences” between the Appellants’ claimed invention and *Corbin*. See Response to Final Office Action, pgs 6-7. Furthermore, the courts have long held that functional language must be accorded patentable weight. See In re Land, 151 USPQ 621 at 635-36 (C.C.P.A., 1966); Intel Corp. v. U.S. Int’l Trade Comm’n, 20 USPQ2d 1161, 1171 (Fed. Cir. 1991).

Accordingly, *Corbin* lacks at least one element of the independent claim 1, as discussed above. As such, *Corbin* does not anticipate the Appellants' invention.

Claims 2 and 7-8 depend, either directly or indirectly, from the independent claim 1. Since *Corbin* does not anticipate Appellants' invention recited in claim 1, *Corbin* fails to anticipate Appellants' invention recited in claims 2 and 7-8.

B. Anticipation of Claims 3-6, 9, and 13 over Corbin

Claims 3-6, 9, and 13 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over *Corbin* on grounds that *Corbin* teaches the claimed subject matter therein.

Corbin is directed to a system for writing cached data to a RAID-5 disk array. In contrast, the Appellants' invention, as recited in independent claim 9, is a system for providing fault tolerant checkpoint data comprising a production server for generating checkpoint data, means for forming distributed checkpoint data comprising subsegments of the checkpoint data, and a plurality of backup servers for storing the distributed checkpoint data, where each of the backup servers in said plurality of backup servers stores at least one subsegment of the distributed checkpoint data.

Corbin is devoid of any teaching or suggestion of the claimed limitation *distributed checkpoint data comprising subsegments of the checkpoint data*. The Examiner contends that the system and RAID-5 disk array in *Corbin* inherently includes creating sub-segments (groups of data) to be spread across the disk drives. First, for the reasons discussed above, the cached data and/or parity in *Corbin* is not checkpoint data. Therefore, *Corbin* does not teach or suggest subsegments of the checkpoint data.

Second, the fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993). "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49

USPQ2d 1949, 1950-51 (Fed. Cir. 1999). "In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." Ex parte Levy, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990).

Based on the above case law, the Examiner has not provided any basis in fact and/or technical reasoning to support the contention that the system in *Corbin* inherently includes creating subsegments of checkpoint data. *Corbin* discloses writing the new data stored in the cache to a RAID-5 disk array, but not creating subsegments of the checkpoint data. Hence, *Corbin* does not inherently disclose the above claimed limitation because the system according to *Corbin* is incapable of creating subsegments of the checkpoint data.

As mentioned above, *Corbin* also fails to teach or suggest the claimed limitations *checkpoint data* and *plurality of backup servers for storing the distributed checkpoint data*. Accordingly, *Corbin* does not teach or suggest each and every limitation as claimed by the Appellants and, as such, does not anticipate the Appellants' invention as recited in independent claim 9.

Claims 3-6, which depend from claim 1, each recite, either directly or indirectly, the feature of subsegmenting the checkpoint data. For the reasons set forth above, Appellants' contend that *Corbin* also fails to anticipate claims 3-6.

Claim 13 depends, either directly or indirectly, from the independent claim 9. Since *Corbin* does not anticipate Appellants' invention recited in claim 9, *Corbin* fails to anticipate Appellants' invention recited in claim 13.

C. Anticipation of Claims 14-16 over Corbin

Claims 14-16 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over *Corbin* on grounds that *Corbin* teaches the claimed subject matter therein.

Corbin is directed to a system for writing cached data to a RAID-5 disk array. In contrast, the Appellants' invention, as recited in independent claim 14, is an apparatus for generating fault tolerant checkpoint data comprising a first server that accesses

checkpoint data, segments the checkpoint data and supplies the segments of checkpoint data to a plurality of second servers.

As mentioned above, *Corbin* does not teach the claimed limitations *checkpoint data, subsegments of checkpoint data or storing distributed checkpoint data amongst a plurality of backup servers*. Independent claim 14 recites similar features, namely a first server that generates checkpoint data, segments the checkpoint data and supplies the segments to a plurality of second servers. Accordingly, *Corbin* does not anticipate the Appellants' invention as recited in independent claim 14.

Claims 15-16 depend, either directly or indirectly, from the independent claim 14. Since *Corbin* does not anticipate Appellants' invention recited in claim 14, *Corbin* fails to anticipate Appellants' invention recited in claims 15-16.

D. Anticipation of Claims 17-19 over Corbin

Claims 17-19 stand rejected under 35 U.S.C. § 102(e) as being unpatentable over *Corbin* on grounds that *Corbin* teaches the claimed subject matter therein.

Corbin is directed to a system for writing cached data to a RAID-5 disk array. In contrast, the Appellants' invention, as recited in independent claim 17, is a method of generating fault tolerant checkpoint data comprising accessing checkpoint data that is produced by a first server, segmenting the checkpoint data and supplying the segments of checkpoint data to a plurality of second servers.

As mentioned above, *Corbin* does not teach the claimed limitations *checkpoint data, segmenting the checkpoint data or supplying the segments of checkpoint data to a plurality of second servers*. Independent claim 17 recites similar features, namely, checkpoint data produced by a first server, segmenting the checkpoint data and supplying the segments to a plurality of second servers. Accordingly, *Corbin* does not anticipate the Appellants' invention as recited in independent claim 17.

Claims 18-19 depend, either directly or indirectly, from the independent claim 17. Since *Corbin* does not anticipate Appellants' invention recited in claim 17, *Corbin* fails to anticipate Appellants' invention recited in claims 18-19.

E. Anticipation of claims 10 and 12 over Corbin

Claims 10 and 12 depend, either directly or indirectly, from the independent claim 9. Because the Appellants contend that *Corbin* fails to anticipate the independent claim 9 for the reasons set forth above, the Appellants further submit that *Corbin* likewise fails to anticipate each of the dependent claims 10 and 12.

In addition, claims 10 and 12 recites features pertaining to the location of the claimed limitation means for forming distributed checkpoint data comprising subsegments of the checkpoint data. To teach such features, the Examiner cited *Corbin*, col. 2, lines 6-8 where “the new parity and the new data is calculated from the old parity and old data...” See final Office Action, p. 4. The cited portion does not teach or suggest the above claimed limitation. A description of calculating new data and new parity does not provide any indication as to the location of the means for forming. As such, *Corbin* does not anticipate Appellants’ claims 10 and 12.

CONCLUSION

For the reasons advanced above, Appellants respectfully urge that the rejections of claims 1-10 and 12-16 as being anticipated under 35 U.S.C. §102(e) is improper. Reversal of the rejections in this appeal is respectfully requested.

Respectfully submitted,

Moser IP Law Group

Date: January 8, 2008

By: /Robert M. Brush/

Robert M. Brush
Registration No. 45,710

MOSE IP LAW GROUP
1030 Broad Street – 2nd Floor
Shrewsbury, NJ 07702
(732) 935-7100
Facsimile: (732) 935-7122
Attorney for Appellant(s)

CLAIMS APPENDIX

1. A method for providing fault tolerant checkpoint data within a server cluster comprising a production server and a plurality of backup servers comprising:
 - accessing checkpoint data within the production server;
 - distributing the checkpoint data; and
 - storing the distributed checkpoint data on the plurality of backup servers.
2. The method of claim 1 wherein the distributing step comprises:
 - creating a redundancy group of checkpoint data; and
 - storing the redundancy group of checkpoint data upon the plurality of backup servers.
3. The method of claim 2 wherein creating the redundancy group comprises:
 - subsegmenting the checkpoint data; and
 - forming groups of subsegments.
4. The method of claim 3 wherein the storing step comprises:
 - striping the subsegments across a plurality of backup servers.
5. The method of claim 2 further comprising:
 - creating parity data for each group.
6. The method of claim 5 wherein the storing step comprises:
 - striping the subsegments and parity data across a plurality of backup servers.
7. The method of claim 1 wherein the storing step comprises:
 - mirroring the checkpoint data onto the plurality of backup servers.
8. The method of claim 1 further comprising:
 - accessing the distributed checkpoint data;
 - reassembling the checkpoint data using the distributed checkpoint data;

using the checkpoint data to initiate execution of software.

9. A system for providing fault tolerant checkpoint data comprising:
 - a production server for generating checkpoint data;
 - means for forming distributed checkpoint data comprising subsegments of the checkpoint data;
 - a plurality of backup servers for storing the distributed checkpoint data, where each of the backup servers in said plurality of backup servers stores at least one subsegment of the distributed checkpoint data.
10. The system of claim 9 wherein the forming means is located within the production server.
12. The system of claim 9 wherein the forming means is located within a computer that is separate from the production server or the plurality of backup servers.
13. The system 6 of claim 9 wherein the means for forming further comprises:
 - means for striping the subsegments onto the plurality of backup servers.
14. Apparatus for generating fault tolerant checkpoint data comprising:
 - a first server that accesses checkpoint data, segments the checkpoint data, and supplies the segments of checkpoint data to a plurality of second servers.
15. The apparatus of claim 14 wherein the first server produces parity data for the segments of checkpoint data and supplies the parity data to the plurality of second servers.
16. The apparatus of claim 14 wherein the segments of check point data are supplied to the plurality of second servers in a striped manner.
17. A method of generating fault tolerant checkpoint data comprising:

accessing checkpoint data that is produced by a first server;
segmenting the checkpoint data; and
supplying the segments of checkpoint data to a plurality of second servers.

18. The method of claim 17 further comprising generating parity data for the segments of checkpoint data and supplying the parity data to the plurality of second servers.

19. The method of claim 17 further comprising supplying the segments of check point data to the plurality of second servers in a striped manner.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE

#70203